

XXXI Всеукраїнська олімпіада з інформатики

Перший тур

А. Піца

Усім учасникам олімпіади відомо, що перед туром слід добре поживитися. З цією метою Степан вирішив замовити вегетаріанську піцу. Піцерія, яку обрав Степан, надає клієнтам можливість обрати інгредієнти для піци самостійно. У наявності піцерії N інгредієнтів, кожен з яких має свою вартість. Степан не може додати до піци один і той самий інгредієнт більше одного разу. Вартість піци рахується як сума вартостей усіх інгредієнтів, що до неї входять.

Неважко підрахувати, що різних варіантів зібрати піцу (окрім нецікавого Степану варіанту піци без інгредієнтів) аж цілих $2^N - 1$. Степан розгубився і вирішив для зручності записати на листочку вартості усіх можливих піц.

Смачно поївши, Степан чудово написав тур олімпіади. Задоволений, він вже збирався йти відпочивати, але зрозумів, що забув вартість кожного з N інгредієнтів. На щастя, Степан зберіг листочок з вартостями усіх варіантів піци, а тому може дізнатися, скільки коштував кожен інгредієнт.

Завдання

Напишіть програму, яка за інформацією про вартості усіх варіантів піци дозволить відновити інформацію про ціни окремих інгредієнтів.

Вхідні дані

В першому рядку вхідного файлу `pizza.in` міститься єдине ціле число N ($1 \leq N \leq 17$) – кількість інгредієнтів. Другий рядок містить рівно $2^N - 1$ цілих додатніх чисел, кожне з яких не перевищує 10^6 – вартості для усіх варіантів піци.

Вихідні дані

В єдиному рядку файлу `pizza.out` має міститися рівно N додатніх цілих чисел у зростаючому порядку – вартості кожного з N інгредієнтів. Якщо існує декілька відповідей, виведіть будь-яку. Якщо Степан щось наплутав і неможливо відновити вартості для кожного інгредієнту, виведіть єдине число `-1`.

Оцінювання

Підзадача	Бали	Додаткові обмеження	Необхідні підзадачі
0	0	Тести з умови	-
1	18	вартості інгредієнтів – попарно різні степені двійки, відповідь завжди існує	-
2	22	$1 \leq N \leq 4$, $1 \leq \text{числа у вхідному файлі} \leq 10$	0
3	22	$1 \leq N \leq 4$	0, 2
4	17	$1 \leq N \leq 10$	0, 2, 3
5	21	Без додаткових обмежень	0, 2, 3, 4

Приклад вхідних та вихідних даних

<code>pizza.in</code>	<code>pizza.out</code>
3	1 2 4
1 2 3 4 5 6 7	
3	1 2 3
1 2 3 3 4 5 6	
2	-1
3 2 2	

В. Маша і мінйони

Мінйони - раса крихітних жовтих створінь. Бути мінйоном це звісно ж круто, але Маша сильніше за них... У розпорядженні у Маші є n мінйонів, кожен з яких характеризується своїми силою та витривалістю.

Філіп попросив Машу передати йому в розпорядження загін мінйонів. Герої вважають, що група мінйонів утворює загін, якщо мінімальне з значень сил цих мінйонів більше або рівне середнього арифметичного їх витривалостей. Також вони вважають, що порожня група мінйонів також утворює загін.

Завдання

Напишіть програму, яка за інформацією про мінйонів, визначить максимальну їх кількість, що утворить загін для Філіпа.

Вхідні дані

Перший рядок файлу `minions.in` містить єдине ціле число n ($1 \leq n \leq 5 \cdot 10^4$) – кількість мінйонів у розпорядженні у Маші. У наступних n рядках знаходяться пари цілих a_i, b_i ($1 \leq a_i, b_i \leq 10^9$) – значення сили та витривалості мінйона з номером i відповідно.

Вихідні дані

Вихідний файл `minions.out` повинен містити єдине ціле число — максимальну кількість мінйонів, які утворюють загін з тих, які є в розпорядженні у Маші.

Оцінювання

Підзадача	Бали	Додаткові обмеження	Необхідні підзадачі
0	0	Тести з умови	-
1	17	$1 \leq n \leq 18$, $1 \leq a_i, b_i \leq 50$	0
2	8	Значення сил усіх мінйонів рівні	-
3	5	Значення витривалостей усіх мінйонів рівні	-
4	16	$1 \leq n \leq 200$	0, 1
5	18	$1 \leq a_i, b_i \leq 50$	0, 1
6	36	Без додаткових обмежень	0, 1, 2, 3, 4, 5

Приклад вхідних та вихідних даних

<code>minions.in</code>	<code>minions.out</code>	Пояснення
3	2	Маша може передати Філіпу загін, що складається з мінйонів з номерами 2 та 3. У такому випадку мінімальне з значень сил цих мінйонів рівне 2, а середнє арифметичне їх витривалостей рівне $(3+1)/2=2$. Група, яка складається з трьох мінйонів не є загonom.
2	0	Неможливо утворити групу, якій належатиме хоча б один мінйон та вона буде загonom, тому відповідь 0.
3 5		
1 4		

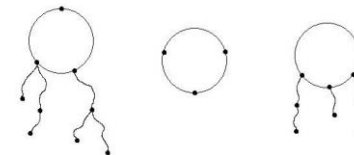
С. Восьминіг

Нещодавно Петрик повернувся додому з відпочинку на морі. Петрик дуже любить тварин, а тому більш за все цей відпочинок запам'ятався йому саме їх великою кількістю. Серед них була і його найулюбленіша — восьминіг.

Повернувшись додому, Петрик одразу ж захотів побудувати багато восьминогів. Для цього він вирішив використовувати свій конструктор. Конструктор являє собою набір гнучких паличок, якими можна з'єднувати диски. Кожен з n дисків має a_i отворів, у які можна вставляти палички. Паличка не може з'єднувати диск з самим собою та будь-яку пару дисків може з'єднувати не більше однієї палички.

В уявленні Петрика, восьминіг виглядає наступним чином

- тіло восьминога складається з трьох або більше дисків, з'єднаних по колу паличками;
- щупальце восьминога складається з декількох дисків, з'єднаних між собою паличками; щупальце може розгалужуватись та не може зростатись;
- кожне щупальце приєднується до якогось диску у тілі восьминога однією паличкою;



Зверніть увагу, що у восьминога може бути декілька щупалець, а може і навпаки не бути жодного щупальця. Більш того, декілька щупалець можуть бути приєднані до одного диска у тілі. На малюнку показані деякі восьминоги.

Завдання

Допоможіть Петрику скласти восьминогів, щоб виконувались наступні умови:

- для побудови були використані усі диски;
- кожен отвір кожного диску був заповнений паличкою;
- кількість побудованих восьминогів була б максимально можливою. Зверніть увагу, що цей пункт є обов'язковим тільки у деяких підзадачах.

Вхідні дані У першому рядку вхідного файлу **octopus.in** задано одне число n ($1 \leq n \leq 10^5$) - кількість дисків у конструкторі Петрика.

У другому рядку записано n цілих чисел a_i ($1 \leq a_i < n$) - кількість отворів у i -му диску.

Третій рядок містить одне ціле число *maximize*, яке дорівнює 0, якщо не треба максимізувати кількість восьминогів, і 1 в противному випадку.

Вихідні дані У перший рядок вихідного файлу **octopus.out** виведіть «Yes», якщо можна побудувати восьминогів, і «No» в противному випадку. У випадку позитивної відповіді виведіть число c - кількість восьминогів. Далі виведіть n рядків. На i -му з них запишіть два цілих числа num_i ($1 \leq num_i \leq c$), p_i - номер восьминога, якому належить диск i , та номер сусіднього диску, який знаходиться ближче до тіла восьминога. Якщо диск i належить тілу, будемо вважати, що $p_i = -1$.

Для кращого розуміння ознайомтесь з прикладами із умови.

Оцінювання

Підзадача	Бали	Додаткові обмеження	Максимізація кількості восьминогів	Необхідні підзадачі
0	0	Тести з умови	Потрібна	-
1	9	Для усіх i виконується $a_i \in \{1, 3\}$	Не потрібна	-
2	13	Для усіх i виконується $a_i \in \{1, 3\}$	Потрібна	1
3	27	-	Не потрібна	1
4	51	-	Потрібна	0, 1, 2, 3

Приклад вхідних та вихідних даних

octopus.in	octopus.out	Пояснення
9 2 1 2 1 1 1 4 3 3 1	Yes 1 1 -1 1 7 1 7 1 3 1 9 1 9 1 -1 1 -1 1 8	Перший приклад з умови відповідає першому восьминогу на малюнку, а другий — другому та третьому.
11 2 2 2 3 3 3 2 2 1 1 1 1	Yes 2 1 -1 1 -1 1 -1 2 -1 2 -1 2 4 2 6 2 7 2 5 2 8	Зверніть увагу, що для даних прикладів можливі і інші схеми, задовольняючі усім необхідним вимогам.
4 2 3 2 3 1	No	

D. Дерево

Добре попрацювавши, Сергій вирішив зайнятись улюбленою справою — малюванням. Звісно, спочатку він намалював Головне Ужляндське Дерево. Це дерево особливе: воно містить n гілок, пронумерованих від 1 до n та розташованих циклічно так, що наступною для першої гілки є друга, для другої - третя, і так далі, а для гілки з номером n - перша.

Початково на кожній з гілок знаходиться деяка кількість пташок (також на гілці може не знаходитись жодна пташка). Всього на дереві m пташок, пронумерованих від 1 до m . Кожна з пташок характеризується своєю вагою: вага пташки з номером i рівна $(a^i + b) \bmod (10^9 + 7)$, де a та b — деякі константи, а $x \bmod y$ позначає остачу від ділення x на y . Відомо, що всі значення ваги пташок попарно різні. Також відомо, що на гілці з номером i початково знаходиться c_i пташок так, що на гілці з номером 1 знаходяться пташки з номерами 1, 2, ..., c_1 , на гілці з номером 2 пташки з номерами $c_1 + 1, c_1 + 2, \dots, c_1 + c_2$ і так далі. Гарантується, що $c_1 + c_2 + \dots + c_n = m$.

Кожної секунди відбувається наступне: одночасно з кожної гілки, на якій знаходиться хоча б одна пташка, перелітає на наступну гілку пташка найменшої ваги. Наприклад, якщо дерево містить 3 гілки, на першій сидять пташки зі значеннями ваги {1, 3, 5}, на другій — {2, 7}, а на третій — {4, 5, 6}, то через секунду значення ваги пташок на гілках будуть рівні {3, 4, 5}, {1, 7}, {2, 5, 6} відповідно. Сергій загадав два числа k та t . Тепер йому цікаво: якою є вага пташки, яка перелітатиме з гілки з номером k у секунду з номером t ?

Завдання

Напишіть програму, яка за інформацією про розміщення пташок на дереві, та числа k та t визначить вагу пташки, яка перелітатиме з гілки з номером k у секунду з номером t ?

Вхідні дані

У першому рядку вхідного файлу **tree.in** задано шість цілих чисел $n m k t a b$ ($1 \leq n \leq 10^4, 1 \leq m \leq 2 \cdot 10^6, 1 \leq k \leq n, 1 \leq t \leq 10^9, 1 \leq a < 10^9 + 7, 0 \leq b < 10^9 + 7$) — кількість гілок дерева, кількість пташок на дереві, числа які загадав Сергій та константи, які визначають вагу пташки відповідно.

У другому рядку задано n цілих чисел c_1, c_2, \dots, c_n ($0 \leq c_i \leq m$), де c_i — початкова кількість пташок на гілці з номером i . Гарантується, що всі значення ваги пташок попарно різні та $c_1 + c_2 + \dots + c_n = m$

Вихідні дані

У вихідний файл **tree.out** виведіть одне ціле число — вагу пташки, яка перелітатиме з гілки з номером k у секунду з номером t , або -1 , якщо перед секундою з номером t на цій гілці не буде жодної пташки.

Оцінювання

Підзадача	Бали	Додаткові обмеження	Необхідні підзадачі
0	0	Тести з умови	-
1	17	$1 \leq n, m, t \leq 100$	0
2	12	$1 \leq n \leq 100, 1 \leq m \leq 50000, 1 \leq t \leq 5000$	0, 1
3	18	$1 \leq n \leq 100, 1 \leq m \leq 50000, 1 \leq t \leq 10^9$	0, 1, 2
4	24	$1 \leq n \leq 10^4, 1 \leq m \leq 50000, 1 \leq t \leq 10^9$	0, 1, 2, 3
5	29	Без додаткових обмежень	0, 1, 2, 3, 4

Приклад вхідних та вихідних даних

tree.in	tree.out
4 9 4 7 2 7	23
3 4 0 2	